Dueling Convex Optimization

Aadirupa Saha [1] **Tomer Koren** [2] **Yishay Mansour** [2]

Abstract

We address the problem of convex optimization with preference (dueling) feedback. Like the traditional optimization objective, the goal is to find the optimal point with the least possible query complexity, however, without the luxury of even a zeroth order feedback. Instead, the learner can only observe a single noisy bit which is win-loss feedback for a pair of queried points based on their function values. The problem is certainly of great practical relevance as in many real-world scenarios, such as recommender systems or learning from customer preferences, where the system feedback is often restricted to just one binary-bit preference information. We consider the problem of online convex optimization (OCO) solely by actively querying $\{0, 1\}$ noisy-comparison feedback of decision point pairs, with the objective of finding a near-optimal point (function minimizer) with the least possible number of queries. For the non-stationary OCO setup, where the underlying convex function may change over time, we prove an impossibility result towards achieving the above objective. We next focus only on the stationary OCO problem, and our main contribution lies in designing a normalized gradient descent based algorithm towards finding a ϵ-best optimal point. Towards this, our algorithm is shown to yield a convergence rate of $\tilde{O}(d\beta/\epsilon\nu^2)$ (ν being the noise parameter) when the underlying function is β-smooth. Further we show an improved convergence rate of just $\tilde{O}(d\beta/\alpha\nu^2 \log \frac{1}{\epsilon})$ when the function is additionally also α-strongly convex.

1. Introduction

Online convex optimization is a very well studied field of research where the goal is to optimize a convex function through sequentially accessing the function information at actively queried points (Flaxman et al., 2005; Ghadimi & Lan, 2013; Bubeck et al., 2017). Due to its great practical relevance in diverse real world scenarios, over the years the problem has been attempted for several problem setups, e.g. optimization with gradient or hessian information (first and second order optimization) (Zinkevich, 2003; Hazan et al., 2007; Hazan & Kale, 2014), function value oracle (zeroth order feedback) (Flaxman et al., 2005; Hazan & Li, 2016; Saha & Tewari, 2011; Yang & Mohri, 2016; Bubeck et al., 2017), multi-point feedback (Ghadimi & Lan, 2013; Shamir, 2017; Agarwal et al., 2010), projection free algorithms (Chen et al., 2018; Sahu et al., 2018) etc., for both adversarial (Abernethy et al., 2008; Dekel et al., 2015; Bach & Perchet, 2016; Bubeck et al., 2015) and stochastic (Agarwal et al., 2011; Wang et al., 2017) settings (Bubeck, 2014; Hazan, 2019; Shalev-Shwartz et al., 2011).

Almost all the existing online optimization literature assumes a first order (gradient information) or at least zeroth order (function value at the queried point) oracle access towards solving the convex optimization problem. However, in reality, many practical applications of online optimization only allow a $\{0, 1\}$-comparison feedback indicating a noisy preference of two (or more) queried points depending on their function values (instead of revealing any information of the gradients at the queried points or even the function values at those points) Such examples are prevalent in many real-world systems that need to collect user preferences instead of their absolute ratings, like recommender systems, online merchandises, search engine optimization, crowdsourcing, drug testing, tournament ranking, social surveys, etc.

This is precisely the reason why there was a massive surge of interest in the bandit community to learn from preference feedback—famously studied as the dueling bandit problem (Komiyama et al., 2015; Ailon et al., 2014; Busa-Fekete & Hüllermeier, 2014; Wu & Liu, 2016). It explicitly models this relative preference information structure, often in the setting of finite action spaces. The goal is to identify the most rewarding activities in hindsight according to a specific score function. Precisely, the learner repeatedly selects a pair of items to be compared to each other in a "duel," and consequently observe a binary stochastic preference feedback of the winning item in this duel. Over the last

[1] Microsoft Research, New York City [2] Blavatnik School of Computer Science, Tel Aviv University, and Google Research Tel Aviv. Correspondence to: Aadirupa Saha <aadirupa.saha@microsoft.com>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

two decades, the bandit literature attempted to study various problems with dueling feedback, but mostly in the stochastic setting with finite arm (decision) spaces. This is since almost all the existing dueling bandit techniques rely on estimating the $K \times K$ preference-matrix and hence the regret scales as $O(K)$. But this, of course, becomes impractical for large (or infinite) action spaces of size K; here lies another primary motivation of our work.

On the other hand, from the point of view of optimization literature, the main reason for the lack of techniques in the comparison based optimization framework is that almost all traditional optimization algorithms require the knowledge (magnitude and direction) of function gradients at the queried points (or at least a noisy estimate of that). However, the gradient-magnitude information is *impossible to obtain* from a binary $0-1$ preference feedback, and thus, the problem is arguably harder than the conventional bandit feedback based optimization framework. Overall, the challenge to work with $\{0,1\}$-preference feedback lies in the inherent disconnect between the feedback observed by the learner and her payoff at any given round; while this disparity already exists in standard dueling bandits even with finite decision space, the setting gets even more challenging for the infinite decision spaces, and additionally with noisy comparison oracles.

Problem Formulation (informal). We address the problem of online convex optimization solely from comparison based oracles. Precisely, given an underlying convex function $f : \mathcal{D} \mapsto [0,1]$ [1] over a convex decision space $\mathcal{D} \subseteq \mathbb{R}^d$, the goal is to find a near-optimal (probably approximately correct) point $\mathbf{x} \in \mathcal{D}$ such that $\mathbf{x}$ satisfies $Pr(f(\mathbf{x}) - f(\mathbf{x}^*) < \epsilon) \geq 1 - \delta$, for any prespecified $\epsilon, \delta \in (0,1)$, by actively querying binary (noisy) comparison feedbacks of the form $\mathbf{1}(f(\mathbf{x}_t) > f(\mathbf{y}_t))$ on a sequence of decision point pairs (aka duels) $\{(\mathbf{x}_t, \mathbf{y}_t)\}_{t=1}^T$. By noisy comparison oracle we appeal to a general setup, where at each round t the true sign feedback $\mathbf{1}(f(\mathbf{x}_t) > f(\mathbf{y}_t))$ could be altered with probability $(1/2 - \nu)$, $\nu \in [0, 0.5]$ being an unknown noise parameter. Given a fixed $\epsilon, \delta \in (0,1)$, the learner's goal is to find such a near-optimal point $\mathbf{x}$ with least possible pairwise-query complexity (T).

The only work that closely attempted a similar problem as described above is (Jamieson et al., 2012). They proposed a coordinate descent based technique to learn from comparison feedback with a provably optimal convergence rate (upto polylogarithmic factors). However, their analysis is only limited to strongly-convex functions, which is a restricted assumption on the function class and precisely, this is why a simple line-search based coordinate descent algorithm works in their case, which is known to fail without strong-convexity. We assume a more general class of only smooth-convex functions, which does not need to be strongly convex, and thus the coordinate descent algorithms do not work in our setup. We instead propose a normalized-gradient descent based algorithm, which is arguably simpler both in terms of implementation and analysis; besides, our convergence rate for strongly convex and smooth functions is order-wise better as their sample complexity incurs extra polylogarithm factors in d, ϵ, ν, which we could get rid of.

Our contributions. The main contributions of this paper can be summarized as follows:

- We formulate the problem of online convex optimization from comparison-based oracles (Sec. 2).
- We first analyze the hardness of the setup under different notion of non-stationarity of the underlying function sequence $(f_1, f_2, \ldots)$ (Thm. 1, Sec. 3).
- Towards designing optimization algorithm, we inferred a descent direction (direction of the gradient at the queried points, aka *normalized gradients*) estimation is enough for our objective. Moreover, when the underlying function is β-smoothly convex, we propose a method to estimate normalized gradient estimates from (noisy) comparison oracles (see Thm. 3 and Lem. 9).
- We propose a *normalized gradient descent* based algorithm for online smooth-convex optimization (Alg. 1) with noisy-comparison oracles. The convergence rate of our algorithm is shown to be $\tilde{O}(d\beta/\epsilon\nu^2)$[2] as derived in Thm. 5 which matches the convergence rates of non-accelarated optimization routines for value based (zeroth order) feedback oracels, which is arguably a easier problem setup that ours (Rem. 2).
- Further when the function is additionally also α-strongly convex, we show an improved convergence rate of just $\tilde{O}(d\beta/\alpha\nu^2 \log \frac{1}{\epsilon})$ as derived in Thm. 7, which is provably optimal in terms of d ν and ϵ (Rem. 3). Overall, we are the first to apply normalized gradient descent based techniques for optimization with preference feedback, consequently our proof techniques are new and involves novelty.

Related work. As motivated above, there has been very little work on convex optimization with preference feedback. (Yue & Joachims, 2009) first address the regret optimization problem for fixed functions f (arm rewards) with preference feedback. However, one of the major differences is that their optimization objective is defined in terms of the 'preferences' (which are directly observable and hence easier to

[1] Note, one can always scale the range of f inside $[0,1]$ as long as f respects a bounded range.

[2] $\tilde{O}(\cdot)$ hides logarithmic dependencies.

optimize) as opposed to defining it w.r.t. f as we considered. Further, their techniques are majorly restricted to the class of 'nice' smooth and differentiable preference functions that allows gradient estimation. This is why their proposed technique could appeal back to the classical gradient descent based optimization algorithms (which are widely studied for the problem of online convex optimization with zeroth and first order oracle (Flaxman et al., 2005; Hazan & Li, 2016; Saha & Tewari, 2011; Yang & Mohri, 2016; Bubeck et al., 2017)), yielding a regret guarantee of $O(\sqrt{dL}T^{3/4})$ with standard analysis (L being the lipschitz parameter of their preference map). Clearly, their algorithm (in fact, any gradient descent based technique) fails in our case, as using only a 1-bit comparison / sign feedback, it is impossible to estimate gradient magnitudes, where lies the main bottleneck of our setup. Consequently, we choose to design normalized gradient descent based optimization algorithms instead.

The only follow up of (Yue & Joachims, 2009) is (Kumagai, 2017), who considers an almost identical problem setup to that of the former and relied on the same gradient descent based ideas to show an improved $O(\sqrt{T})$ regret guarantee, but at the cost of even stricter assumptions on the preference models: Precisely it requires the underlying function f to be twice continuously differentiable, L-Lipschitz, strongly convex and smooth and a thrice differentiable, rotation-symmetric preference map. Their setting is hence far from our current optimization objective and does not apply to our 1-bit noisy comparison feedback model, which is not even singly differentiable.

As described above, the work most related to our setup is by (Jamieson et al., 2012) which, however, only applies to the specific class of α-strongly β-smooth convex functions. On the other hand, our algorithm applies to a more general class of β-smooth convex functions, simpler to implement, requires newer insights for the performance analysis, and also respects better convergence rates (see Rem. 3 for details).

2. Preliminaries and Problem Statement

Notation. Let $[n] = \{1, 2, \ldots n\}$, for any $n \in \mathbb{N}$. Given a set S, for any two items $x, y \in S$, we denote by $x \succ y$ the event x is preferred over y. For any $r > 0$, let $\mathcal{B}_d(r)$ and $\mathcal{S}_d(r)$ denote the ball and the surface of the sphere of radius r in d dimensions respectively. $\mathbf{I}_d$ denotes the $d \times d$ identity matrix. For any vector $\mathbf{x} \in \mathbb{R}^d$, $\|\mathbf{x}\|_2$ denotes the ℓ_2 norm of vector $\mathbf{x}$. $\mathbf{1}(\varphi)$ is generically used to denote an indicator variable that takes the value 1 if the predicate φ is true, and 0 otherwise. $\text{sign}(x) = +1$ if $x \geq 0$ or -1 otherwise, $\forall x \in \mathbb{R}$. Unif(S) denotes a uniform distribution over any set S.

Problem Setup. We solve the online convex optimization problem with 0-1 preference feedback. In the most general setup, the environment chooses a sequence of convex functions $f_1, f_2, \ldots$ (unknown to the learner), such that $f_t : \mathbb{R}^d \mapsto \mathbb{R}$ for all t. At any iteration, the goal of the learner is to pick a pair of points $(\mathbf{x}_t, \mathbf{y}_t)$ upon which it gets to see a binary $0-1$ bit noisy comparison feedback o_t s.t.:

$$Pr(o_t = \mathbf{1}(f_t(\mathbf{y}_t) > f_t(\mathbf{x}_t))) = 1/2 + \nu,$$

where $\nu \in (0, 1/2]$ is the (unknown) noise-parameter.

Based on the variability in the function sequence $f_1, f_2, \ldots$, one can categorize the problem into the following cases:

1. *Stationary*: In this case the functions are assumed to be fixed across time, i.e. $f_t = f$ for some unknown but fixed function $f : \mathbb{R}^d \mapsto \mathbb{R}$.
2. *Stochastic*: In this case the functions are drawn from some (unknown) stochastic distribution $\mathcal{P}$ from a given (known) class of convex functions $\mathcal{F} \subseteq \{f : \mathbb{R}^d \mapsto \mathbb{R}\}$, i.e. at any time time $t \in [T]$, $f_t \sim \mathcal{P}$.
3. *Adversarial*: The most general setup for any arbitrary sequence of convex functions $\{f_t\}_{t=1}^T$. Note this case can be seen as a special case of *Stochastic* setup where $\mathcal{F} = \{f_t\}_{t=1}^T$, and $Pr(f_t = f_i) = 0$ for all $i \neq t$.

Objective. We consider the standard (ϵ, δ)-probably approximately correct function optimization objective: Given any fixed choice of $\epsilon, \delta \in (0, 1)$, the objective of the learner is to find a decision point $\mathbf{x} \in \mathbb{R}^d$ with minimum possible pairwise-query complexity $\{(\mathbf{x}_t, \mathbf{y}_t)_{t=1}^T\}$ such that the final point $\mathbf{x}$ must satisfy:

$$Pr(\bar{f}(\mathbf{x}) - \bar{f}(\mathbf{x}^*) < \epsilon) \geq 1 - \delta,$$

where $\bar{f}(x) := \sum_{t=1}^T \frac{\mathbb{E}_{f_t \sim \mathcal{P}}[f_t(x)]}{T}$, $\forall \mathbf{x} \in \mathbb{R}^d$ denotes the average function value, and $\mathbf{x}^* := \arg\min_{\mathbf{z} \in \mathbb{R}^d} \bar{f}(z)$ denotes the minimizer of $\bar{f}$.

3. Non-stationarity leads to Impossibility

We first analyze the fundamental performance limit of the problem for non-stationary sequence of functions, precisely the *Stochastic* and *Adversarial* setp. The main result of this section (Thm. 1) shows the optimization problem is in fact impossible to solve in both cases. This is precisely because a comparison based oracle does not take into account the scale of the function values, neither the learner has a way to estimate this from the comparison feedback. Thus in the non-stationary setup it is always possible problem instances to make a decision point appear to be the best in terms of the observed preferences, where as in reality it is not even an ϵ-optimal point compared to $\mathbf{x}^*$. The proof of Thm. 1 gives a formal argument.

Theorem 1. *For the setup of Stochastic and Adversarial it is always possible to construct problem instances where the optimization problem becomes infeasible (impossible to identify the true minimizer $\mathbf{x}^*$).*

Proof. **Instance $\mathcal{I}$:** Consider a simple problem instance $\mathcal{I}$ for *Stochastic* setup with decision set $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2\}$, function class $\mathcal{F} = \{f_1, f_2\}$, and assume $\mathcal{P}$ is such that:

$$f_t = \begin{cases} f_1, & \text{with probability } 0.99 \\ f_2, & \text{otherwise} \end{cases},$$

where f_1 and f_2 is respectively defined as:

$$f_1(\mathbf{x}) = \begin{cases} 0.01, & \text{if } \mathbf{x} = \mathbf{x}_1 \\ 0, & \text{otherwise} \end{cases}, \quad f_2(\mathbf{x}) = \begin{cases} 0, & \text{if } \mathbf{x} = \mathbf{x}_1 \\ 1, & \text{otherwise} \end{cases}.$$

Then note that at any round t, $Pr_{f_t \sim \mathcal{P}}(\mathbf{x}_2 \succ \mathbf{x}_1) = Pr(f_t = f_1) \cdot Pr(\mathbf{x}_2 \succ \mathbf{x}_1 \mid f_t = f_1) + Pr(f_t = f_2) \cdot Pr(\mathbf{x}_2 \succ \mathbf{x}_1 \mid f_t = f_2) = 0.99 \cdot 1 + 0.01 \cdot 0 = 0.99$. Thus on expectation, $\mathbf{x}_2$ wins over $\mathbf{x}_1$ almost always (99 times out of 100 duels).

However, on the other hand, in terms of the function values $\mathbf{x}_1$ is the optimal point (minimizer of $\bar{f}$, see Objective in Sec. 2). This can be easily inferred just by noting $\mathbb{E}_{f_t \sim \mathcal{P}}[f_t(\mathbf{x}_1)] = Pr(f_t = f_1) \cdot f_t(\mathbf{x}_1) + Pr(f_t = f_2) \cdot f_t(\mathbf{x}_1) = 0.99 \cdot 0.01 + 0.01 \cdot 0 = 0.0099 < 0.01 = 0.01 \cdot 1 = \mathbb{E}_{f_t \sim \mathcal{P}}[f_t(\mathbf{x}_2)]$. So for the instance $\mathcal{I}$, we have $\mathbf{x}^* = 1$.

Instance $\mathcal{I}'$: Now let us consider a slightly tweaked version of instance $\mathcal{I}$, say $\mathcal{I}'$ which has the exactly same $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2\}$ and a slightly different function class $\mathcal{F} = \{f_1, f_2'\}$, where f_2' is defined as: $f_2'(\mathbf{x}) = \begin{cases} 0, & \text{if } \mathbf{x} = \mathbf{x}_1 \\ 0.1, & \text{otherwise} \end{cases}$.

Now note, even for $\mathcal{I}'$ at any round t, we still see $Pr_{f_t \sim \mathcal{P}}(\mathbf{x}_2 \succ \mathbf{x}_1) = Pr(f_t = f_1) \cdot Pr(\mathbf{x}_2 \succ \mathbf{x}_1 \mid f_t = f_1) + Pr(f_t = f_2') \cdot Pr(\mathbf{x}_2 \succ \mathbf{x}_1 \mid f_t = f_2') = 0.99 \cdot 1 + 0.01 \cdot 0 = 0.99$.

The interesting thing however is for this case, in terms of the function values $\mathbf{x}_2$ is indeed the optimal point (i.e. $\mathbf{x}^* = \mathbf{x}_2$). This follows by noting $\mathbb{E}_{f_t \sim \mathcal{P}}[f_t(\mathbf{x}_1)] = Pr(f_t = f_1) \cdot f_t(\mathbf{x}_1) + Pr(f_t = f_2') \cdot f_t(\mathbf{x}_1) = 0.99 \cdot 0.01 + 0.01 \cdot 0 = 0.0099 > 0.001 = 0.01 \cdot 0.1 = \mathbb{E}_{f_t \sim \mathcal{P}}[f_t(\mathbf{x}_2)]$. So for the instance $\mathcal{I}'$, we have $\mathbf{x}^* = 2$.

Now consider any $\epsilon < \min(0.01 - 0.0099, 0.0099 - 0.001) = 0.0001$. The only ϵ-optimal arm for $\mathcal{I}$ is $\mathbf{x}_1$, whereas for $\mathcal{I}'$ is $\mathbf{x}_2$. But in both case learner observes the exactly same preference feedback, i.e. $Pr(\mathbf{x}_2 \succ \mathbf{x}_1) = 0.99$. No it really has no ways to distinguish $\mathcal{I}$ from $\mathcal{I}'$ and consequently identify the ϵ-best optimal point of the true underlying instance. Note the dispute arises since the binary comparison based preference feedback reveals no information of the magnitude of the underlying function values, and any learning algorithm would observe $\mathbf{x}_2$ beats $\mathbf{x}_1$ almost always (with 0.99 probability), irrespective of if the true optimal arm is $\mathbf{x}_1$ (i.e. true instance is $\mathcal{I}$) or $\mathbf{x}_2$ (i.e. the true instance is $\mathcal{I}'$). □

Remark 1. Above example also proves the impossibility result for the *Adversarial* setup as the *Stochastic* setup is just a simple and special case of the former.

4. Estimating Descent Direction (Normalized Gradient) using Comparison Oracle

As motivated in Sec. 1, one of our main contribution lies in the analysis to obtain an unbiased estimate of normalized-gradient at any desired point $\mathbf{x} \in \mathbb{R}^d$ from 1-bit comparison feedback. Thm. 3 shows the main result of this section, but before that we find it useful to introduce the following key lemma that shows how one can obtain an unbiased estimate of the direction of any vector $\mathbf{g} \in \mathbb{R}^d$, i.e., its normalized version $\frac{\mathbf{g}}{\|\mathbf{g}\|}$, using only a 1-bit comparison (sign function) oracle. Following is a general result whose scope lies beyond our specific problem setup.

Lemma 2. *For a given vector $\mathbf{g} \in \mathbb{R}^d$ and a random unit vector $\mathbf{u}$ drawn uniformly from $\mathcal{S}_d(1)$, we have*

$$\mathbb{E}[\text{sign}(\mathbf{g} \cdot \mathbf{u})\mathbf{u}] = \frac{c}{\sqrt{d}} \frac{\mathbf{g}}{\|\mathbf{g}\|},$$

for some universal constant $c \in [\frac{1}{20}, 1]$.

Proof sketch. Without loss of generality we can assume $\|\mathbf{g}\| = 1$, since normalizing $\mathbf{g}$ does not affect the left-hand side. First, let us show that $\mathbb{E}[\text{sign}(\mathbf{g} \cdot \mathbf{u})\mathbf{u}] = \gamma \mathbf{g}$ for some $\gamma \in \mathbb{R}$. Consider the reflection matrix along $\mathbf{g}$ given by $P = 2\mathbf{g}\mathbf{g}^\top - I$, and examine the random vector $\mathbf{u}' = P\mathbf{u}$. Observe that $\text{sign}(\mathbf{g} \cdot \mathbf{u}') = \text{sign}\left(2\|\mathbf{g}\|^2 \mathbf{g}^\top \mathbf{u} - \mathbf{g}^\top \mathbf{u}\right) = \text{sign}(\mathbf{g} \cdot \mathbf{u})$. Since $\mathbf{u}'$ is also a random vector on the unit sphere, we have

$$\begin{aligned} \mathbb{E}[\text{sign}(\mathbf{g} \cdot \mathbf{u})\mathbf{u}] &= \tfrac{1}{2}\mathbb{E}[\text{sign}(\mathbf{g} \cdot \mathbf{u})\mathbf{u}] + \tfrac{1}{2}\mathbb{E}[\text{sign}(\mathbf{g} \cdot \mathbf{u}')\mathbf{u}'] \\ &= \tfrac{1}{2}\mathbb{E}[\text{sign}(\mathbf{g} \cdot \mathbf{u})\mathbf{u}] + \tfrac{1}{2}\mathbb{E}[\text{sign}(\mathbf{g} \cdot \mathbf{u})(2\mathbf{g}\mathbf{g}^\top - \mathbf{I})\mathbf{u}] \\ &= \mathbb{E}[(\mathbf{g} \cdot \mathbf{u})\,\text{sign}(\mathbf{g} \cdot \mathbf{u})]\mathbf{g}. \end{aligned}$$

Thus, $\mathbb{E}[\text{sign}(\mathbf{g} \cdot \mathbf{u})\mathbf{u}] = \gamma \mathbf{g}$ for $\gamma = \mathbb{E}[|\mathbf{g} \cdot \mathbf{u}|]$.

It remains to bound γ, which by rotation invariance equals $\mathbb{E}[|u_1|]$. For an upper bound, observe that by symmetry $\mathbb{E}[u_1^2] = \frac{1}{d}\mathbb{E}[\sum_{i=1}^d u_i^2] = \frac{1}{d}$ and thus $\mathbb{E}[|u_1|] \le \sqrt{\mathbb{E}[u_1^2]} =$

$\frac{1}{\sqrt{d}}$. We turn to prove a lower bound on γ. If $\mathbf{u}$ were a Gaussian random vector with i.i.d. entries $u_i \sim \mathcal{N}(0, 1/d)$, then from standard properties of the (truncated) Gaussian distribution we would have gotten that $\mathbb{E}[|u_1|] = \sqrt{2/\pi d}$. For $\mathbf{u}$ uniformly distributed on the unit sphere, u_i is distributed as $v_1/\left\|\mathbf{v}\right\|$ where $\mathbf{v}$ is Gaussian with i.i.d. entries $\mathcal{N}(0, 1/d)$. We then can write

$$\begin{aligned}\Pr\left(|u_1| \geq \frac{\epsilon}{\sqrt{d}}\right) &= \Pr\left(\frac{|v_1|}{\|\mathbf{v}\|} \geq \frac{\epsilon}{\sqrt{d}}\right) \\ &\geq 1 - \Pr\left(|v_1| < \frac{1}{\sqrt{d}}\right) - \Pr\left(\|\mathbf{v}\| > \frac{1}{\epsilon}\right).\end{aligned}$$

Since $\sqrt{d}v_1$ is a standard Normal, we have

$$\begin{aligned}\Pr\left(|v_1| < \frac{1}{\sqrt{d}}\right) &= \Pr\left(-1 < \sqrt{d}v_1 < 1\right) \\ &= 2\Phi(1) - 1 \leq 0.7,\end{aligned}$$

and since $\mathbb{E}[\left\|\mathbf{v}\right\|^2] = 1$ an application of Markov's inequality gives $\Pr\left(\|\mathbf{v}\| > \frac{1}{\epsilon}\right) = \Pr\left(\|\mathbf{v}\|^2 > \frac{1}{\epsilon^2}\right) \leq \epsilon^2\mathbb{E}[\left\|\mathbf{v}\right\|^2] = \epsilon^2$. For $\epsilon = \frac{1}{4}$ this implies that $\Pr\left(|u_1| \geq 1/4\sqrt{d}\right) \geq \frac{1}{5}$, whence $\gamma = \mathbb{E}[|u_1|] \geq 1/20\sqrt{d}$. □

Using above result we obtain the main result of this section. The complete proof is moved to Appendix. A.

Theorem 3. *If f is β-smooth, for any $\mathbf{u} \sim$ Unif$(\mathcal{S}_d(1))$, $\delta \in (0, 1)$ and vector $\mathbf{b} \in \mathcal{S}_d(1)$:*

$$\begin{aligned}&\mathbb{E}_{\mathbf{u}}[\text{sign}(f(\mathbf{x} + \delta\mathbf{u}) - f(\mathbf{x} - \delta\mathbf{u}))\mathbf{u}^\top\mathbf{b}] \\ &\quad \leq \frac{c}{\sqrt{d}}\frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|} + 2\lambda,\end{aligned}$$

for some universal constant $c \in [\frac{1}{20}, 1]$, and $\lambda = \frac{3\beta\delta}{\|\nabla f(\mathbf{x})\|}\sqrt{d\log\frac{\|\nabla f(\mathbf{x})\|}{\sqrt{d}\beta\delta}}$.

Proof sketch. The proof mainly lies on the following lemma that shows how to the comparison feedback of two close points–$\mathbf{x} + \gamma\mathbf{u}$ and $\mathbf{x} - \gamma\mathbf{u}$–can be used to recover directional information of the gradient of f at point $\mathbf{x}$.

Lemma 4. *If f is β-smooth, for any $\mathbf{u} \sim$ Unif$(\mathcal{S}_d(1))$, and $\gamma \in (0, 1)$, then with probability at least $1 - \lambda$ where $\lambda = \frac{3\beta\gamma}{\|\nabla f(\mathbf{x})\|}\sqrt{d\log\frac{\|\nabla f(\mathbf{x})\|}{\sqrt{d}\beta\gamma}}$ we have*

$$\text{sign}(f(\mathbf{x} + \gamma\mathbf{u}) - f(\mathbf{x} - \gamma\mathbf{u}))\mathbf{u} = \text{sign}(\nabla f(\mathbf{x}) \cdot \mathbf{u})\mathbf{u}.$$

Consequently, for any vector $\mathbf{b} \in \mathcal{S}_d(1)$ we have $\left|\mathbb{E}_{\mathbf{u}}[\text{sign}(f(\mathbf{x}+\gamma\mathbf{u}) - f(\mathbf{x}-\gamma\mathbf{u}))\mathbf{u}^\top\mathbf{b}] - \mathbb{E}_{\mathbf{u}}[\text{sign}(\nabla f(\mathbf{x}) \cdot \mathbf{u})\mathbf{u}^\top\mathbf{b}]\right| \leq 2\lambda.$

We give a brief outline of the above proof, the complete analysis could be found in Appendix A. From smoothness we have $|f(\mathbf{x} + \gamma\mathbf{u}) - f(\mathbf{x} - \gamma\mathbf{u}) - 2\gamma\mathbf{u} \cdot \nabla f(\mathbf{x})| \leq \beta\gamma^2$. Therefore, if $\beta\gamma^2 \leq \gamma\left|\mathbf{u} \cdot \nabla f(\mathbf{x})\right|$, we will have that $\text{sign}(f(\mathbf{x} + \gamma\mathbf{u}) - f(\mathbf{x} - \gamma\mathbf{u})) = \text{sign}(\mathbf{u} \cdot \nabla f(\mathbf{x}))$. Let us analyse $\Pr_{\mathbf{u}}(\beta\gamma \geq |\mathbf{u} \cdot \nabla f(\mathbf{x})|)$. We know for $\mathbf{v} \sim \mathcal{N}(\mathbf{0}_d, \mathcal{I}_d)$, $\mathbf{u} := \mathbf{v}/\|\mathbf{v}\|$ is uniformly distributed on the unit sphere. Then we show its possible to write: $\mathbf{P}_{\mathbf{u}}\left(|\mathbf{u} \cdot \nabla f(\mathbf{x})| \leq \beta\gamma\right) \leq \mathbf{P}_{\mathbf{v}}\left(|\mathbf{v} \cdot \nabla f(\mathbf{x})| \leq 2\beta\gamma\sqrt{d\log(1/\gamma')}\right) + \gamma'$.

Again since $\mathbf{v} \cdot \nabla f(\mathbf{x}) \sim \mathcal{N}(0, \|\nabla f(\mathbf{x})\|^2)$, for any $\gamma > 0$ that $\Pr\left(|\mathbf{v} \cdot \nabla f(\mathbf{x})| \leq \gamma\right) \leq \frac{2\gamma}{\|\nabla f(\mathbf{x})\|\sqrt{2\pi}} \leq \frac{\gamma}{\|\nabla f(\mathbf{x})\|}$. Combining the inequalities, we have that $\text{sign}(f(\mathbf{x} + \gamma\mathbf{u}) - f(\mathbf{x} - \gamma\mathbf{u})) = \text{sign}(\mathbf{u} \cdot \nabla f(\mathbf{x}))$ except with probability at most

$$\begin{aligned}&\inf_{\gamma'>0}\left\{\gamma' + \frac{2\beta\gamma\sqrt{d\log(1/\gamma')}}{\|\nabla f(\mathbf{x})\|}\right\} \\ &\leq \frac{3\beta\gamma}{\|\nabla f(\mathbf{x})\|}\sqrt{d\log\frac{\|\nabla f(\mathbf{x})\|}{\sqrt{d}\beta\gamma}} = \lambda\end{aligned}$$

As for the claim about the expectation, note that for any vector $\mathbf{b} \in \mathcal{S}_d(1)$, $\left|\mathbb{E}_{\mathbf{u}}[\text{sign}(f(\mathbf{x}+\gamma\mathbf{u}) - f(\mathbf{x}-\gamma\mathbf{u}))\mathbf{u}^\top\mathbf{b}] - \mathbb{E}_{\mathbf{u}}[\text{sign}(\nabla f(\mathbf{x}) \cdot \mathbf{u})\mathbf{u}^\top\mathbf{b}]\right| \leq 2\lambda$, since with probability $1 - \lambda$ the two expectations are identical, and otherwise, they differ by at most 2. This concludes the proof of Lem. 4.

The result of Thm. 3 now simply follows by combining the guarantees of Lem. 2 and 4. □

5. Noiseless case: Analysis for *Sign-Feedback* with Normalized-Gradient Descent

In this section, we analyse the case of 'no-noise', i.e. $\nu = 1/2$ (see the comparison feedback model in Sec. 2), i.e. upon querying any duel $(\mathbf{x}, \mathbf{y})$ the learner gets access to its true sign feedback $\mathbf{1}(f(\mathbf{x}) > f(\mathbf{y}))$. We start by presenting our main algorithm (Alg. 1) for β-smooth convex optimization which is based on the technique of normalized gradient descent which appeals back to our results derived in Sec. 4. We next analyse its rate of convergence which respects a PAC sample complexity bound of $O\left(\frac{d\beta\|x_1 - \mathbf{x}^*\|^2}{\epsilon}\right)$ (Thm.

5), $\mathbf{x}_1$ being the initial point considered by the algorithm. Following this, we also address the setup for combined α-strongly convex, β-smooth functions and show that for this case one can achieve an improved convergence rate with by simply using a blackbox routine for beta-smooth optimization (say our Alg. 1) iteratively. Precisely, our resulting algorithm (Alg. 2) is shown to yield a convergence rate of $O\Big(\frac{d\beta}{\alpha}\log\frac{\alpha}{\epsilon}\Big)$ (Thm. 7), and this respects the optimal convergence rates in terms of ϵ and d (Rem. 3).

5.1. f is β-smooth convex

Algorithmic ideas. We start by recalling that, from a single 0-1 bit comparison feedback, one can not hope to recover the gradient estimates. This is since a comparison oracle does not reveal any information on the scale (magnitude) of the function values (see Sec. 3 for a motivating example). So the traditional gradient descent based techniques are bound to fail in our case in the first place. However, in Sec. 4 we show, if not the entire gradient, we can almost recover the direction of the gradient (aka normalized gradient) at any desired point (see Thm. 3). Thus we appeal to the technique of *Normalized Gradient Descent* to solve the current problem.

Starting from an initial point $\mathbf{x}_1$, the algorithm sequentially goes on finding a nearly unbiased normalized gradient estimate $h_t := \text{sign}(f(\mathbf{x}_t') - f(\mathbf{y}_t'))\mathbf{u}_t$,[3] $\mathbf{u}_t \sim \text{Unif}(\mathcal{S}_d(1))$ being any random unit norm d-dimensional vector, and take η-sized steps along the negative direction of the estimated normalized gradients. Also note, at each time t, we keep track of the 'so-far-minimum' point $\tilde{\mathbf{x}}_t$ (current best). Essentially at all t, $\tilde{\mathbf{x}}_t$ traces $\arg\min_{\tau=1}^{t} f(\mathbf{x}_\tau)$. This is unavoidable since without the knowledge of the gradient magnitude the algorithm does not have a way to gauge whether $\mathbf{x}_t$ is very close or too far form the optimal point $\mathbf{x}^*$.

Theorem 5 (Noiseless-Optimization: β-smooth function (Alg. 1))**.** *Consider f to be β smooth. Suppose Alg. 1 is run with $\eta = \frac{\sqrt{\epsilon}}{20\sqrt{d\beta}}$, $\gamma = \frac{(\epsilon/\beta)^{3/2}}{240\sqrt{2}d(D+\eta T)^2\sqrt{\log 480\sqrt{\beta d}(D+\eta T)/\sqrt{2\epsilon}}}$ and $T_\epsilon = O\Big(\frac{d\beta D}{\epsilon}\Big)$, where $D \geq \|\mathbf{x}_1 - \mathbf{x}^*\|^2$ (is an assumed known upper bound). Then Alg. 1 returns $\mathbb{E}[f(\tilde{\mathbf{x}}_{T+1})] - f(\mathbf{x}^*) \leq \epsilon$ with sample complexity $2T_\epsilon$.*

Remark 2. The convergence rate in Thm. 5 is same as that achieved by non-accelerated optimization algorithms for smooth convex optimization with zeroth-order feedback which reveals the true function value at queried the points (e.g., Nesterov & Spokoiny, 2017). This is arguably a richer feedback model compared to our 1-bit comparison oracle, as one can always obtain the comparison feedback from zeroth order oracle but not the other way. So a comparison based optimization is always as hard as that of the zeroth order optimization problem, as we can always solve the later problem, given a black-box for the first.

Algorithm 1 β-*NGD*$(\mathbf{x}_1, \eta, \gamma, T_\epsilon)$

```
Input: Initial point: $\mathbf{x}_1 \in \mathbb{R}^d$ such that $D := \|\mathbf{x}_1 - \mathbf{x}^*\|^2$ (assume known), Learning rate $\eta$, Perturbation parameter $\gamma$, Query budget $T_\epsilon$
(Recall the desired error tolerance is $\epsilon > 0$)
Initialize Current minimum $\tilde{\mathbf{x}}_1 \in \mathbb{R}^d$
for $t = 1, 2, 3, \ldots, T_\epsilon$ do
    Sample $\mathbf{u}_t \sim \text{Unif}(\mathcal{S}_d(1))$
    $\mathbf{x}_t' := \mathbf{x}_t + \gamma\mathbf{u}_t$
    $\mathbf{y}_t' := \mathbf{x}_t - \gamma\mathbf{u}_t$
    Play the duel $(\mathbf{x}_t', \mathbf{y}_t')$, and receive binary preference feedback $o_t = \mathbf{1}(f(\mathbf{x}_t') < f(\mathbf{y}_t'))$. Set $o_t' = 2o_t - 1$.
    Update $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta\mathbf{h}_t$, where $\mathbf{h}_t = o_t'\mathbf{u}_t$
    Query the pair $(\mathbf{x}_{t+1}, \tilde{\mathbf{x}}_t)$.
    Update $\tilde{\mathbf{x}}_{t+1} \leftarrow \begin{cases} \mathbf{x}_{t+1} & \text{if } o_t' = -1 \\ \tilde{\mathbf{x}}_t & \text{otherwise } (o_t' = +1) \end{cases}$
end for
Return $\tilde{\mathbf{x}}_{T+1}$
```

Proof sketch (Thm. 5). Consider the following cases:

Case 1. $(f(\mathbf{x}_0) - f(\mathbf{x}^*) \leq \epsilon)$**:** In this case, the initial point $\mathbf{x}_1$ is already good enough (close enough to $\mathbf{x}^*$).

Case 2. $(f(\mathbf{x}_0) - f(\mathbf{x}^*) > \epsilon)$**:** In this case we appeal to Lem. 6 which ensures finding a point $\mathbf{x}_t$ for $t \in [T_\epsilon]$ such that $\mathbb{E}[f(\mathbf{x}_{t+1})] - f(\mathbf{x}^*) \leq \epsilon$.

The bound of Thm. 5 now follows noting that by definition $\tilde{\mathbf{x}}_{t+1} = \min(\mathbf{x}_1, \ldots, \mathbf{x}_t)$ so as long as $\exists t \in [T_\epsilon]$ with $\mathbb{E}[f(\mathbf{x}_{t+1})] - f(\mathbf{x}^*) \leq \epsilon$, we have $\mathbb{E}[f(\tilde{\mathbf{x}}_{T+1})] - f(\mathbf{x}^*) \leq \epsilon$.

Finally the sample complexity bound follows straightforwardly since Alg. 1 takes T_ϵ as input and for each $t \in [T_\epsilon]$ it makes 2 pairwise comparisons, respectively $(\mathbf{x}_t', \mathbf{y}_t')$ and $(\mathbf{x}_{t+1}, \tilde{\mathbf{x}}_t)$, making the total query complexity $2T_\epsilon$.

The rest of the proof we briefly sketch the proof of the following main lemma:

Lemma 6. *Consider f is β smooth. In Alg. 1, if the initial point $\mathbf{x}_1$ is such that $f(\mathbf{x}_1) - f(\mathbf{x}^*) > \epsilon$, and given any $\epsilon > 0$ the parameters T_ϵ, γ and η is as defined in Thm. 5. Then there exists at least one t such that $\mathbb{E}[f(\mathbf{x}_{t+1})] - f(\mathbf{x}^*) \leq \epsilon$, i.e. $\min_{t\in[T]} \mathbb{E}[f(\mathbf{x}_{t+1})] - f(\mathbf{x}^*) \leq \epsilon$.*

Consider any $t = 1, 2, \ldots T$, such that $f(\mathbf{x}_\tau) > f(\mathbf{x}^*) + \epsilon$, $\forall \tau \in [t]$, and we denote by $\mathbf{n}_t = \frac{\nabla f(\mathbf{x}_t)}{\|\nabla f(\mathbf{x}_t)\|}$. Let $\mathbf{y}_t := \mathbf{x}^* + \sqrt{\frac{2\epsilon}{\beta}}\mathbf{n}_t$. Then using β-smoothness of f, $f(\mathbf{y}_t) \leq f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)(\mathbf{y}_t - \mathbf{x}^*) + \frac{\beta}{2}\|\mathbf{y}_t - \mathbf{x}^*\|^2 = f(\mathbf{x}^*) + \epsilon$.

[3] We take cues form the result of Thm. 3 or more precisely Lem. 4 to come up with the functional form of h_t.

Thus we conclude $f(\mathbf{y}_t) < f(\mathbf{x}_t)$. From Lem. 14 we get $\mathbf{n}_t^\top(\mathbf{y}_t - \mathbf{x}_t) \le 0 \implies \mathbf{n}_t^\top\left(\mathbf{x}^* + \sqrt{\frac{2\epsilon}{\beta}}\mathbf{n}_t - \mathbf{x}_t\right) \le 0 \implies -\mathbf{n}_t^\top(\mathbf{x}_t - \mathbf{x}^*) \le -\sqrt{\frac{2\epsilon}{\beta}}$.

Observation 1. Note for any $t \in [T]$, $f(\mathbf{x}_t) > f(\mathbf{x}^*) + \epsilon$ implies $\|\mathbf{x}_t - \mathbf{x}^*\| > \sqrt{\frac{2\epsilon}{\beta}}$ (since from β-smoothness of f we know that $f(\mathbf{x}_t) - f(\mathbf{x}^*) \le \frac{\beta}{2}\|\mathbf{x}_t - \mathbf{x}^*\|^2$.

Observation 2. Since $f(\mathbf{y}_t) < f(\mathbf{x}_t)$, from properties of convex function (see Lem. 14) in Appendix B we get $\mathbf{n}_t^\top(\mathbf{y}_t - \mathbf{x}_t) \le 0 \implies \mathbf{n}_t^\top\left(\mathbf{x}^* + \sqrt{\frac{2\epsilon}{\beta}}\mathbf{n}_t - \mathbf{x}_t\right) \le 0$

We denote by $\mathcal{H}_t$ the history $\{\mathbf{x}_\tau, \mathbf{u}_\tau\}_{\tau=1}^{t-1} \cup \mathbf{x}_t$ till time t. Then note that by the update rule of $\mathbf{x}_{t+1}$ (and since $\|\mathbf{h}_t\| = 1$):

$$\|\mathbf{x}_{t+1} - \mathbf{x}^*\|^2 \le \|\mathbf{x}_t - \mathbf{x}^*\|^2 - 2\eta\mathbf{h}_t^\top(\mathbf{x}_t - \mathbf{x}^*) + \eta^2$$

Now starting from the above equation along with a series of inference steps stemming from Observations 1,2, Thm. 3, some properties of the convex functions, and appropriate choices of γ and η, it can be shown that above implies: $\mathbb{E}_{\mathcal{H}_t}[\mathbb{E}_{\mathbf{u}_t}[\|\mathbf{x}_{t+1} - \mathbf{x}^*\|^2 \mid \mathcal{H}_t]] \le \mathbb{E}_{\mathcal{H}_t}[\|\mathbf{x}_t - \mathbf{x}^*\|^2] - \frac{(\sqrt{2}-1)\epsilon}{400d\beta}$. Now summing over $t = 1, \ldots T$ and further applying laws of iterated expectation, this finally boils down to: $\mathbb{E}_{\mathcal{H}_T}[\|\mathbf{x}_{T+1} - \mathbf{x}^*\|^2] \le \|\mathbf{x}_1 - \mathbf{x}^*\|^2 - \frac{(\sqrt{2}-1)\epsilon T}{400d\beta}$.

Thus, we see that, if indeed $f(\mathbf{x}_\tau) - f(\mathbf{x}^*) > \epsilon$ continues to hold for all $\tau = 1, 2, \ldots T$, then $\mathbb{E}[\|\mathbf{x}_{T+1} - \mathbf{x}^*\|^2] \le 0$, for $T \ge \frac{400d\beta}{(\sqrt{2}-1)\epsilon}(\|\mathbf{x}_1 - \mathbf{x}^*\|^2)$, which basically implies $\mathbf{x}_{T+1} = \mathbf{x}^*$ (i.e. $f(\mathbf{x}_{T+1}) = f(\mathbf{x}^*)$). Otherwise there must have been a time $t \in [T]$ such that $f(\mathbf{x}_t) - f(\mathbf{x}^*) < \epsilon$. This concludes the proof with $T_\epsilon = T$.

This concludes the proof of Lem. 6, as well as the proof of Thm. 5 which was already proved earlier (assuming the result of Lem. 6 holds good). □

5.2. f is α-strongly convex and β-smooth

We next show a better convergence rate for α-strongly convex and β-smooth function. For this case, we note that one can simply reuse any optimal optimization algorithm for β-smooth convex functions (we use our Alg. 1) as a blackbox to design an optimal algorithm for α-strongly convex, β-smooth functions. Our proposed method (α, β)-*NGD* (Alg. 2) adapts a phase-wise iterative optimization approach, where inside each phase we use the Alg. 1 as a blackbox to locate a ϵ_k-optimal point in that phase with exponentially decaying ϵ_k (with $\epsilon_1 = 1$) and warm start the $(k+1)$-th phase from the optimizer returned by β-*NGD* in the k-th phase. The method works precisely due to the nice properly of strong-convex functions where nearness in function values implies nearness from the optimal $\mathbf{x}^*$ in ℓ_2-norm (see Lem. 8 and proof of Thm. 7). The algorithm is described in Alg. 2.

Algorithm 2 (α, β)-*NGD*(ϵ)

```
Input: Error tolerance $\epsilon > 0$
Initialize Initial point: $\mathbf{x}_1 \in \mathbb{R}^d$ such that $D := \|\mathbf{x}_0 - \mathbf{x}^*\|^2$ (assume known).
Phase counts $k_\epsilon := \lceil\log_2\left(\frac{\alpha}{\epsilon}\right)\rceil$, $t \leftarrow \frac{800d\beta}{(\sqrt{2}-1)\alpha}$
$\eta_1 \leftarrow \frac{\sqrt{\epsilon_1}}{20\sqrt{d\beta}}, \epsilon_1 = \frac{400d\beta D}{(\sqrt{2}-1)t_1} = 1, t_1 = t\|\mathbf{x}_1 - \mathbf{x}^*\|^2$
$\gamma_1 \leftarrow \frac{(\epsilon_1/\beta)^{3/2}}{240\sqrt{2}d(D+\eta_1 t_1)^2\sqrt{\log 480\sqrt{\beta d}(D+\eta_1 t_1)/\sqrt{2\epsilon_1}}}$.
Update $\mathbf{x}_2 \leftarrow \beta$-NGD$\big(x_1, \eta_1, \gamma_1, t_1\big)$
for $k = 2, 3, \ldots, k_\epsilon$ do
    $\eta_k \leftarrow \frac{\sqrt{\epsilon_k}}{20\sqrt{d\beta}}, \epsilon_k = \frac{400d\beta}{(\sqrt{2}-1)t_k}, t_k = 2t$
    $\gamma_k \leftarrow \frac{(\epsilon_k/\beta)^{3/2}}{240\sqrt{2}d(1+\eta_k t_k)^2\sqrt{\log 480\sqrt{\beta d}(1+\eta_k t_k)/\sqrt{2\epsilon_k}}}$.
    Update $\mathbf{x}_{k+1} \leftarrow \beta$-NGD$\big(x_k, \eta_k, \gamma_k, t_k\big)$
end for
Return $\tilde{\mathbf{x}} = \mathbf{x}_{k_\epsilon+1}$
```

Theorem 7 (Noiseless-Optimization: α-strongly convex and β-smooth case (Alg. 2))**.** *Consider f to be α-strongly convex and β-smooth. Then Alg. 2 returns $\mathbb{E}[f(\tilde{\mathbf{x}})] - f(\mathbf{x}^*) \le \epsilon$ with sample complexity (number of pairwise comparisons) $O\Big(\frac{d\beta}{\alpha}\big(\log_2\left(\frac{\alpha}{\epsilon}\right) + \|\mathbf{x}_1 - \mathbf{x}^*\|^2\big)\Big)$.*

Remark 3. The line search algorithm proposed by (Jamieson et al., 2012) also achieves the same convergence rate for strongly convex functions, modulo some additional multiplicative polylogarithmic terms (in d, ϵ, ν) in the sample complexity bounds, which we could get rid of. Their lower bound justifies the tightness of the analysis of Thm. 7 in terms of the problem parameters d, ν and ϵ. However, it is important to note that our algorithm is much simpler both in implementation and analysis. Besides, our proposed methods are certainly more general that applies to the class of non-strongly convex functions as well (see Thm. 5), where (Jamieson et al., 2012) fails.

Proof sketch (Thm. 7). We start by recalling an important property of strongly convex function (proof in Appendix B):

Lemma 8. *If $f : \mathbb{R} \mapsto \mathbb{R}$ is an α-strongly convex function, with $\mathbf{x}^*$ being the minimizer of f. Then for any $\mathbf{x} \in \mathcal{R}$, $\frac{\alpha}{2}\|\mathbf{x}^* - \mathbf{x}\|^2 \le f(\mathbf{x}) - f(\mathbf{x}^*)$.*

Also let $\mathcal{H}_k := \{\mathbf{x}_{k'}, (\mathbf{x}_{t'}, \mathbf{y}_{t'}, o_{t'})_{t' \in t_{k'}}\}_{k'=1}^k \cup \{\mathbf{x}_{k+1}\}$ denotes the complete history till the end of phase k for all $k \in [k_\epsilon]$.

By Thm. 5, for any fixed $T > 0$, when β-*NGD* (Alg. 1) is run with $\eta = \left(\frac{\sqrt{\epsilon}}{20\sqrt{d\beta}}\right)$, $\gamma =$

$\frac{(\epsilon/\beta)^{3/2}}{240\sqrt{2}d(D+\eta T)^2\sqrt{\log{480\sqrt{\beta d}(D+\eta T)/\sqrt{2\epsilon}}}}$ and $\epsilon = \frac{400d\beta D}{(\sqrt{2}-1)T}$, where $D := \|\mathbf{x}_1 - \mathbf{x}^*\|^2$. Then Alg. 1 returns $\mathbb{E}[f(\tilde{\mathbf{x}}_{T+1})] - f(\mathbf{x}^*) \leq \epsilon = \frac{400d\beta\|\mathbf{x}_1 - \mathbf{x}^*\|^2}{(\sqrt{2}-1)T}$ with sample complexity $2T$. However, in this case since f is also α-strongly convex, using Lem. 8 we get:

$$\mathbb{E}[{}^{\alpha}/{}_{2}\|\tilde{\mathbf{x}}_{T+1} - \mathbf{x}^*\|^2] \leq \frac{400d\beta\|\mathbf{x}_1 - \mathbf{x}^*\|^2}{(\sqrt{2}-1)T} \qquad (1)$$

i.e $\mathbb{E}[\|\tilde{\mathbf{x}}_{T+1} - \mathbf{x}^*\|^2] \leq \frac{800d\beta\|\mathbf{x}_1 - \mathbf{x}^*\|^2}{(\sqrt{2}-1)\alpha T}$. Now initially for $k = 1$, clearly applying the above result for $T = t\|\mathbf{x}_1 - \mathbf{x}^*\|^2$, we get $\mathbb{E}[\|\mathbf{x}_2 - \mathbf{x}^*\|^2] \leq \frac{800d\beta\|\mathbf{x}_1 - \mathbf{x}^*\|^2}{(\sqrt{2}-1)\alpha T} = 1$

Thus, for any $k = 2, \ldots k_\epsilon$, given the initial point $\mathbf{x}_k$, if we run β-*NGD* with $T = 2t = \frac{1600d\beta}{(\sqrt{2}-1)\alpha}$, we get from (1): $\mathbb{E}_{\mathcal{H}_k}[\|\tilde{\mathbf{x}}_{k+1} - \mathbf{x}^*\|^2 \mid \mathcal{H}_{k-1}] \leq \frac{800d\beta\|\mathbf{x}_k - \mathbf{x}^*\|^2}{(\sqrt{2}-1)\alpha T} = \frac{\|\mathbf{x}_k - \mathbf{x}^*\|^2}{2}$. This implies given the history till phase $k-1$, using (1) and our choice of t_k,

$$\begin{aligned}
&\mathbb{E}_{\mathcal{H}_k}[f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*) \mid \mathcal{H}_{k-1}] \\
&\quad\leq \mathbb{E}_{\mathcal{H}_k}[\frac{1}{4\alpha}\|\mathbf{x}_k - \mathbf{x}^*\|^2 \mid \mathcal{H}_{k-1}] \\
&\quad\leq \frac{1}{4\alpha}(\frac{1}{2})^{k-1}\|\mathbf{x}_1 - \mathbf{x}^*\|^2 \leq \frac{1}{\alpha 2^{k+1}}.
\end{aligned}$$

Thus, to ensure at $k = k_\epsilon$, $\mathbb{E}[f(\mathbf{x}_{k_\epsilon+1}) - f(\mathbf{x}^*)] \leq \epsilon$, this demands $({}^1/{}_2)^{k_\epsilon+1}\alpha \leq \epsilon$, or equivalently $\frac{\alpha}{2\epsilon} \leq 2^{k_\epsilon+1}$, which justifies the choice of $k_\epsilon = \log_2\left(\frac{\alpha}{\epsilon}\right)$. By Thm. 5, recall running the subroutine β-*NGD*$(\mathbf{x}_{k-1}, \eta_k, \gamma_k, t_k)$ actually requires a query complexity of $2t_k$, and hence the total query complexity of Alg. 2 becomes $4tk_\epsilon + t_1 = O\Big(\frac{800d\beta}{(\sqrt{2}-1)\alpha}(\log_2\left(\frac{\alpha}{\epsilon}\right) + \|\mathbf{x}_1 - \mathbf{x}^*\|^2)\Big)$. □

6. General Case: Noisy-*Sign-Feedback*

Recall from Sec. 2 that in the most general setup, we consider a noisy comparison feedback such that $Pr(o_t = \mathbf{1}(f_t(\mathbf{y}_t) > f_t(\mathbf{x}_t))) = {}^1/{}_2 + \nu$, for some $\nu \in (0, 0.5]$. Our proposed algorithms in the earlier section require the knowledge of true sign feedback $o_t = \mathbf{1}(f(\mathbf{x}_t) > f(\mathbf{y}_t))$ for every pairwise query $(\mathbf{x}_t, \mathbf{y}_t)$, but in reality the comparison oracle could be noisy and return incorrect signs where they fail. We resolve the problem by a 'resampling-trick' described below.

6.1. `sign-recovery`: De-noising the Oracle

Main idea of `sign-recovery` (Alg. 3). We note that, given any pair $(\mathbf{x}, \mathbf{y})$, by re-querying it for $O(\frac{1}{\nu^2}\log\frac{1}{\nu^2\delta})$ times, one can obtain the true the sign feedback $\mathbf{1}(f(\mathbf{x}) > f(\mathbf{y}))$ with high probability.

Lemma 9. *For any dueling pair $(\mathbf{x}, \mathbf{y})$ and confidence parameter $\delta \in (0, 1]$, with probability at least $(1 - \delta/2)$, the output o of `sign-recovery`$(\mathbf{x}, \mathbf{y}, \delta)$ (Alg. 3) returns the true indicator value of $\mathbf{1}(f(\mathbf{x}) > f(\mathbf{y}))$ with at most $t = O\big(\frac{1}{\nu^2}\log\frac{1}{\nu^2\delta}\big)$ pairwise queries. More precisely:*

$$\begin{aligned}
&Pr\Big(o = \mathbf{1}(f(\mathbf{x}) - f(\mathbf{y})) \text{ and } t = O\big(\frac{1}{\nu^2}\log\frac{1}{\nu^2\delta}\big)\Big) \\
&\quad > 1 - \frac{\delta}{2},
\end{aligned}$$

where the probability is taken on randomness of the observed comparison sequence $\{o_\tau\}_{\tau\in[t]}$.

Remark 4. Note the algorithm does not require the knowledge of the noise parameter ν.

Algorithm 3 `sign-recovery`$(\mathbf{x}, \mathbf{y}, \delta)$

```
1:  Input: Dueling pair: $(\mathbf{x}, \mathbf{y})$. Desired confidence $\delta \in [0, 1]$. Initialize $w \leftarrow 0$
2:  for t = 1,2, . . . do
3:      Play $(\mathbf{x}, \mathbf{y})$.
4:      Receive $o_t \leftarrow$ noisy-preference$\big(\mathbf{1}(f(\mathbf{x}) < f(\mathbf{y}))\big)$
5:      Update $w \leftarrow w + o_t$, $p_t(\mathbf{x}, \mathbf{y}) \leftarrow \frac{w}{t}$.
6:      $\texttt{conf}_t := \sqrt{\frac{\log(8t^2/\delta)}{2t}}$
7:      $l_t(\mathbf{x}, \mathbf{y}) := p_t(\mathbf{x}, \mathbf{y}) - \texttt{conf}_t$
8:      $l_t(\mathbf{y}, \mathbf{x}) := 1 - p_t(\mathbf{x}, \mathbf{y}) - \texttt{conf}_t$
9:      if either $l_t(\mathbf{x}, \mathbf{y}) > 1/2$ or $l_t(\mathbf{y}, \mathbf{x}) > 1/2$: Break.
10: end for
11: Compute $o \leftarrow \begin{cases} 1 & \text{if } l_t(\mathbf{x}, \mathbf{y}) > 1/2 \\ 0 & \text{otherwise} \end{cases}$
12: Return $o$
```

Proof sketch Let Alg. 3 stops at round τ. The proof uses Hoeffding's inequality which ensures at all iteration $t \in [\tau]$, with probability at least $(1 - \delta)$, it satisfies $|p_t(\mathbf{x}, \mathbf{y}) - Pr(\{s = 1\})| \leq \texttt{conf}_t$. (See Lem. 16 and 17 in Appendix C.) Note this equivalently implies $|p_t(\mathbf{y}, \mathbf{x}) - Pr(\{s = 0\})| \leq \texttt{conf}_t$, where $p_t(\mathbf{y}, \mathbf{x}) = 1 - p_t(\mathbf{x}, \mathbf{y})$ and since $Pr(\{s = 0\}) = 1 - Pr(\{s = 1\})$. We now consider the following cases:

Case 1. $\mathbf{1}(f(\mathbf{x}) < f(\mathbf{y})) = 1$: So in this case, at any $t \in [\tau]$, $Pr(o_t = 1) = Pr(s = 1) = 1/2 + \nu$. Then $l_t(\mathbf{x}, \mathbf{y}) \geq p_t(\mathbf{x}, \mathbf{y}) - \texttt{conf}_t \geq Pr(o_t = 1) - 2\texttt{conf}_t = 1/2 + \nu - 2\texttt{conf}_t$. So we get $l_t(\mathbf{x}, \mathbf{y}) > 1/2$ whenever $2\texttt{conf}_t < \nu$, or $t > \frac{2}{\nu^2}\log\frac{8t^2}{\delta}$. Note that later is satisfied for any $t \geq \frac{8}{\nu^2}\log\frac{64}{\nu^2\delta}$. This can be easily verified setting $a = 2/\nu^2$ and $b = 8/\delta$ in Lem. 18.

Then assuming the confidence bounds of Lem. 17 holds good at all t, we can safely conclude that the algorithm satisfies the stopping criterion (see Line #9 in Alg. 3) for $\tau = O\big(\frac{1}{\nu^2}\log\frac{1}{\nu^2\delta}\big)$ This implies the correctness and sample complexity of `sign-recovery`.

Case 2. $\mathbf{1}(f(\mathbf{x}) < f(\mathbf{y})) = 0$: In this case $Pr(o_t = 0) = 1/2 + \nu$. Then $l_t(\mathbf{y}, \mathbf{x}) \geq p_t(\mathbf{y}, \mathbf{x}) - \texttt{conf}_t \geq Pr(o_t =$

$0) - 2\texttt{conf}_t = 1/2 + \nu - 2\texttt{conf}_t$. The rest follows same as Case 1. (complete proof in Appendix C). □.

6.2. Algorithms and Analysis

β-smooth convex functions. We can essentially re-use Alg. 1 again, except since we don't have access to the true comparison $\mathbf{1}(f(\mathbf{x}'_t) > f(\mathbf{y}'_t))$, but only a noisy 1-bit feedback of the former, we need to estimate the true $\texttt{sign}$ with high probability. For this, we appeal to our `sign-recovery` subroutine to obtain the true o_t, as required in the Line #7 and #9 of Alg. 1. For completeness, the pseudocode of *Robust-β-NGD* (Alg. 4) is given in Appendix C.

Theorem 10 (Noisy-Optimization: Smooth Convex functions)**.** *Consider f to be β smooth. Suppose β-NGD (Alg. 4) is run with $\eta = \Big(\frac{\sqrt{\epsilon}(2p-1)^2}{20\sqrt{d\beta}}\Big)$, $\gamma = \frac{(\epsilon/\beta)^{3/2}}{240\sqrt{2}d(D+\eta T)^2\sqrt{\log 480\sqrt{\beta d}(D+\eta T)/\sqrt{2\epsilon}}}$ and $T_\epsilon = O\Big(\frac{d\beta D}{\epsilon}\Big)$, where $D := \|\mathbf{x}_1 - \mathbf{x}^*\|^2$. Then, for any given $\delta \in (0,1)$, with high probability at least $(1-\delta)$ (over the randomness of noisy comparison feedback o_t) it returns $\mathbb{E}[f(\tilde{\mathbf{x}}_{T+1})] - f(\mathbf{x}^*) \le \epsilon$ with sample complexity $O\Big(\frac{d\beta D}{\epsilon\nu^2}\log\frac{d\beta D}{\epsilon\nu^2\delta}\Big)$ (expectation is taken over randomness of the algorithm).*

Proof sketch. The proof precisely follows from the proof of Thm. 5, as Lem. 9 ensures at every round o_t gets assigned to the true $\mathbf{1}(f(\mathbf{x}_t) < f(\mathbf{y}_t))$ with 'sufficiently' high probability. This ensures the correctness of the algorithm. The sample complexity simply follows by taking into account the additional $\tilde{O}\big(\frac{1}{\nu^2}\log\frac{1}{\nu^2}\big)$ pairwise-queries incurred in `sign-recovery` subroutine (per dueling-pair) to recover the correct comparison feedback at each round. □

α-strongly convex and β-smooth functions. In this case again, we can reuse our Alg. 2, originally proposed for the noiseless setup. To accommodate the noisy comparison-feedback oracle, in this case it requires to use the robust version of the underlying blackbox algorithm, i.e. *Robust-β-NGD* in Line #3 and #6 of Alg. 2. For completeness, the full algorithm (Alg. 5) is given in Appendix C.

Theorem 11 (Noisy-Optimization: Strongly Convex and Smooth case)**.** *Let f is α-strongly convex and β-smooth. Then given any $\delta \in (0,1)$, with probability at least $(1-\delta)$ (over randomness of noisy comparison feedback o_t), Alg. 5 with using Robust-β-NGD (Alg. 4) as the underlying blackbox), returns $\mathbb{E}[f(\tilde{\mathbf{x}})] - f(\mathbf{x}^*) \le \epsilon$ in sample complexity $O\Big(\frac{d\beta}{\nu^2\alpha}(\log_2\big(\frac{\alpha}{\epsilon}\big) + D)\Big)\log\frac{d\beta D\log(\alpha/\epsilon)}{\nu^2\delta}$ (expectation is taken over randomness of the algorithm), where $D := \|\mathbf{x}_1 - \mathbf{x}^*\|^2$.*

7. Conclusion and Perspective

We address the problem of online convex optimization with comparison feedback and design normalized gradient descent based algorithms that yield fast convergence guarantees for smooth convex and strongly convex+smooth functions. Moving forward, there are many open questions to address, including unifying the class of preference maps (that maps a duel-score to preference feedback), analyze the regret minimization objective, understanding information-theoretic performance limits, or even generalizing the framework to general subsetwise preference-based learning problem. The setup of optimization from preference feedback being relatively new and almost unexplored, the scopes of potential future directions are vast.

Acknowledgments

The work of YM has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 882396), by the Israel Science Foundation (grant number 993/17) and the Yandex Initiative for Machine Learning at Tel Aviv University. TK was supported in part by the Israeli Science Foundation (ISF) grant no. 2549/19, by the Len Blavatnik and the Blavatnik Family foundation, and by the Yandex Initiative in Machine Learning.

References

Abernethy, J. D., Hazan, E., and Rakhlin, A. Competing in the dark: An efficient algorithm for bandit linear optimization. *Conference on Learning Theory*, 2008.

Agarwal, A., Dekel, O., and Xiao, L. Optimal algorithms for online convex optimization with multi-point bandit feedback. In *COLT*, pp. 28–40. Citeseer, 2010.

Agarwal, A., Foster, D. P., Hsu, D. J., Kakade, S. M., and Rakhlin, A. Stochastic convex optimization with bandit feedback. In *Advances in Neural Information Processing Systems*, pp. 1035–1043, 2011.

Ailon, N., Karnin, Z. S., and Joachims, T. Reducing dueling bandits to cardinal bandits. In *ICML*, volume 32, pp. 856–864, 2014.

Bach, F. and Perchet, V. Highly-smooth zero-th order online optimization. In *Conference on Learning Theory*, pp. 257–283, 2016.

Bubeck, S. Convex optimization: Algorithms and complexity. *arXiv preprint arXiv:1405.4980*, 2014.

Bubeck, S., Dekel, O., Koren, T., and Peres, Y. Bandit convex optimization:\sqrtt regret in one dimension. In *Conference on Learning Theory*, pp. 266–278, 2015.

Bubeck, S., Lee, Y. T., and Eldan, R. Kernel-based methods for bandit convex optimization. In *Proceedings of* *the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 72–85. ACM, 2017.

Busa-Fekete, R. and Hüllermeier, E. A survey of preference-based online learning with bandit algorithms. In *International Conference on Algorithmic Learning Theory*, pp. 18–39. Springer, 2014.

Chen, L., Zhang, M., and Karbasi, A. Projection-free bandit convex optimization. *arXiv preprint arXiv:1805.07474*, 2018.

Dekel, O., Eldan, R., and Koren, T. Bandit smooth convex optimization: Improving the bias-variance tradeoff. In *Advances in Neural Information Processing Systems*, pp. 2926–2934, 2015.

Flaxman, A. D., Kalai, A. T., and McMahan, H. B. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 385–394. Society for Industrial and Applied Mathematics, 2005.

Ghadimi, S. and Lan, G. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Hazan, E. Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*, 2019.

Hazan, E. and Kale, S. Beyond the regret minimization barrier: optimal algorithms for stochastic strongly-convex optimization. *The Journal of Machine Learning Research*, 15(1):2489–2512, 2014.

Hazan, E. and Li, Y. An optimal algorithm for bandit convex optimization. *arXiv preprint arXiv:1603.04350*, 2016.

Hazan, E., Agarwal, A., and Kale, S. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

Jamieson, K. G., Nowak, R., and Recht, B. Query complexity of derivative-free optimization. In *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2012.

Komiyama, J., Honda, J., Kashima, H., and Nakagawa, H. Regret lower bound and optimal algorithm in dueling bandit problem. In *Conference on Learning Theory*, pp. 1141–1154, 2015.

Kumagai, W. Regret analysis for continuous dueling bandit. In *Advances in Neural Information Processing Systems*, pp. 1489–1498, 2017.

Nesterov, Y. and Spokoiny, V. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.

Saha, A. and Tewari, A. Improved regret guarantees for online smooth convex optimization with bandit feedback. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 636–642, 2011.

Sahu, A. K., Zaheer, M., and Kar, S. Towards gradient free and projection free stochastic optimization. *arXiv preprint arXiv:1810.03233*, 2018.

Shalev-Shwartz, S. et al. Online learning and online convex optimization. *Foundations and trends in Machine Learning*, 4(2):107–194, 2011.

Shamir, O. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *Journal of Machine Learning Research*, 18(52):1–11, 2017.

Wang, Y., Du, S., Balakrishnan, S., and Singh, A. Stochastic zeroth-order optimization in high dimensions. *arXiv preprint arXiv:1710.10551*, 2017.

Wu, H. and Liu, X. Double Thompson sampling for dueling bandits. In *Advances in Neural Information Processing Systems*, pp. 649–657, 2016.

Yang, S. and Mohri, M. Optimistic bandit convex optimization. In *Advances in Neural Information Processing Systems*, pp. 2297–2305, 2016.

Yue, Y. and Joachims, T. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 1201–1208, 2009.

Zinkevich, M. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 928–936, 2003.